Aeon Graph Network (AGN)

A Decentralized Open-Weight Network over a Verifiable Associative Memory
Proof-of-work from an energy law; supply bounded by forgetting

Marcel Langjahr

Pantareon
Pantareon Foundations

langjahr@pantareon.io · https://pantareon.io/

July 2026

Abstract

We present *Aeon Graph Network* (AGN), a protocol aimed at a decentralized open-weight network with two halves: a fixed, permissively licensed open-weight model as the reasoning layer, and a shared living information graph as the memory and knowledge layer beneath it. The graph is the part that is novel and the part this paper reports on. Its shared state is a single graph in which storing, associating, retrieving and forgetting are one physical process, and what the network sells is that process — not access to the model, whose weights anyone can also run locally. Retrieval is relaxation of an energy functional Φ to a coalition; learning is the trace that settling leaves on the edges; forgetting is resonance decay below a floor. Because Φ is a Lyapunov function, monotone descent is a cheap, unforgeable certificate that work was really done — the proof-of-work is the substrate's own energy law rather than an unrelated puzzle. Because information that is not retrieved is pruned and its bonded stake burned, the monetary supply is bounded by *forgetting* rather than by a fixed cap.

The engine is integer fixed-point (Q32.32) throughout, so the whole consensus-critical path is bit-reproducible. That contract is now tested rather than assumed: the canonical state hash and region root, together with the raw limbs of the transcendental and division primitives, are pinned as literal reference vectors and reproduce byte-identically across five targets spanning architecture, word size, byte order, OS and ABI — x86-64/Windows, aarch64 on both Linux and Apple Silicon, 32-bit i686 (where the 128-bit multiply path is lowered to software routines) and big-endian s390x. All but the last run on native hardware, and CI re-checks every one of them on each push.

On that substrate we report measured, control-backed capability. Associative capacity is 8 patterns at N=48 (Hopfield reference ≈7); a covariance write rule lifts overlapping-pattern recall from 10% to 52%, and both storage and prototype recall rise together with scale (25%→79% and 33%→72% across N=48 . . . 192). Directed sequence, conditional branching on shared states (50%→100%) and self-selected context depth are demonstrated with the controls that fail. Analogy is carried from a given correspondence to one discovered jointly with the mapping (97%) and finally to inference: relations never observed are generated correctly 83% of the time against a 7% control. Several objects coexist in shared cells at distinct phases and are demultiplexed at 100% precision, where a single-phase control collapses to 44%. Deterministic forgetting matches its closed form to 2.3×10^{-10}.

The coordination layer — ledger, BFT finality, governance, verified sync, peer discovery — is built and gated, but has never been run across machines, and nothing is yet economically at stake. We state that boundary explicitly rather than blur it, and describe what closing it requires.

Availability of Source Code: The protocol is licensed under Apache-2.0. That is a design requirement rather than a concession: a system whose whole premise is that anonymous third parties host and extend it cannot be licensed per operator, and a permissionless network is legally impossible without such a licence. The repository is currently private and will be opened when the network strand is ready for third-party operators; access can be granted before then upon reasonable request for technical due diligence, subject to a non-disclosure agreement. Pantareon builds and maintains the protocol; commercial arrangements concern the application layer above it, integration, and funded co-development.

Scope of this Paper: This is a technical status report intended for partners and technically liter-

ate readers; it is not submitted for peer review. It reports what is built and measured today, and separates that cleanly from what is designed but unproven. All figures come from deterministic, reproducible runs with the controls stated; every result cited is reproduced by a named gate in the repository, and the two canonical hashes are checked on every push by CI across the five targets above. There is no live network, no token with value, and no user funds at risk.

1 Introduction

Decentralized-AI proposals to date decentralize *existing* machine learning. Miners run conventional training or inference, validators score the output, and a chain settles payment. The architecture of the system being served is untouched: data and training remain separate phases, storage grows monotonically and is replicated forever, and the network's guarantee is agreement among scorers rather than proof that the computation was honest.

AGN starts from a different premise. Its substrate is an energy-based associative memory in the Hopfield lineage, generalised to a sparse graph with complex-valued units: a unit of stored information is a node, an association is an edge, a query is a clamp, retrieval is letting the local subgraph descend an energy functional into a settled coalition, and learning is the trace that settling leaves on the edges. There is no separate training corpus and no separate index, because storage, indexing and learning are the same object.

The consequential choice is that the dynamics are *dissipative* rather than conservative. The substrate is not asked to preserve a quantity; it is asked to descend one, monotonically, to a fixed point. That is what turns a retrieval into a receipt, and it is the property the rest of this paper rests on.

Three consequences follow from the construction rather than being engineered on top.

The proof of work is the physics. A relaxation episode is a deterministic function of (region state, clamp, seed, config). Along an honest trajectory Φ is non-increasing and reaches a fixed point. A fabricated answer has no valid descending trajectory, and a descending trajectory cannot be cheaply forged because every step is fixed by the previous state and the local subgraph. Verification is therefore recomputation of a few sampled steps, not a separate puzzle.

The supply is bounded by forgetting. Each node carries a resonance that decays continuously and is replenished only by retrieval. Below a floor the node is pruned and the stake bonded to it is burned. Minting tracks retained information and burning tracks forgotten information, so the canonical state is self-limiting — the state-bloat problem that no UTXO or account chain solves is answered by the substrate's own metabolism rather than by policy.

Determinism is a protocol requirement, not a preference. Consensus rests on independent parties deriving identical roots from identical inputs. The engine is therefore integer fixed-point end to end, including the transcendentals; there is no floating point anywhere in the consensus-critical path.

Two clarifications about scope, since they determine how the rest should be read. The graph is a memory, not a reasoner, and the protocol does not wait for reasoning to emerge from it: the target architecture binds a fixed open-weight model at the protocol boundary as the reasoning layer (§6), with the graph as the memory and knowledge layer underneath. And the memory half is the half that needs a protocol at all — the model's weights are free and reproducible by anyone, while a shared, permissionless, self-pruning memory is not. Everything measured in this paper is the memory half; the model integration is designed and not yet built, and we mark that line rather than blur it.

2 Architecture

AGN is three layers, and conflating the first two is the central modelling error to avoid.

Layer 0 — the information substrate. The graph of information units: Φ-relaxation, plasticity, resonance, pruning. Cooperative, dissipative energy minimisation; every node descends the same functional, which is a Lyapunov function, so the system provably converges. Economy-free.

Layer 1 — the miner graph. Hosts, stake, validation and payment. This layer is *adversarial*: participants may cheat, so there is no shared energy an attacker descends voluntarily and Layer 1 cannot be a Lyapunov system. It is secured by proof-of-relaxation, stake and slashing instead.

The relation between them is exact rather than metaphorical: Layer 0 is partitioned into regions, each held by one miner, and Layer 1 is precisely the quotient graph obtained by contracting each region to a node. The core function is literally the subgraph inside a miner.

Layer 2 — the trading layer. A non-custodial onramp, an application above the protocol. It may be central; Layers 0 and 1 must not be.

One firewall runs through the design: *resonance*

is not stake. Resonance is driven solely by usage and retrieval; a miner earns in proportion to the resonance it serves but can never raise that resonance by staking. Without this, capital would decide what is true.

2.1 The energy functional

Node state is a complex unit $z_i = a_i e^{i\theta_i}$: activation as magnitude, binding phase as argument. The functional

$$\Phi(z) = -\sum_{(i,j)\in E} w_{ij}\,\mathrm{Re}(\bar{z}_i z_j) + \sum_i \Lambda(a_i) + \frac{\beta}{2}\Big(\sum_i a_i^2 - K\Big)^2 + \frac{\gamma}{2}\sum_{i\neq j}|z_i z_j|^2$$

has four terms: phase-gated association, an activation cost, a sparsity budget that makes nodes compete so only a small coalition can be active, and clamp fidelity. Symmetric couplings make Φ a Lyapunov function under gradient flow; the descent is the certificate.

Three timescales separate cleanly: relaxation of z (retrieval, per query), plasticity of w (consolidation), and resonance decay with pruning (the lifecycle). The separation is what keeps the per-episode descent guarantee intact while the structure learns and the graph forgets.

3 The Arithmetic Contract

Everything above is void if two honest participants on different hardware compute different roots from the same block. This is the load-bearing assumption of the whole network strand, and it is the one most often left untested: a harness that compares a release build against a debug build on one machine is a codegen test wearing a hardware test's name, and one that compares runs only against *each other* will report success even when every build is uniformly wrong.

We therefore pin literal reference vectors: the canonical fast-state hash `864fb0cd1f8d498f`, the canonical region root `4e3b32d3...`, and the raw Q32.32 limbs of `exp`, `cos_sin`, `sqrt`, `mul` and `div`. The primitives are pinned separately on purpose: an end-to-end hash tells you *that* something diverged, the limbs tell you *where*.

target	word	bytes	host	result
x86-64 Windows/MSVC	64	LE	native	PASS
aarch64 Linux/GNU	64	LE	native	PASS
aarch64 macOS	64	LE	native	PASS
i686 Linux/GNU	**32**	LE	native	PASS
s390x Linux/GNU	64	**BE**	emulated	PASS

Table 1: Reference vectors, byte-identical across architecture, word size, byte order, OS and ABI. Re-checked by CI on every push.

Two of the five carry most of the information. On the 32-bit target the 128-bit multiply and divide used by every fixed-point operation are lowered to compiler-builtin software routines rather than native instructions, so identical limbs mean the contract survives a genuinely different lowering of its hottest operation. On the big-endian target, every commitment in the protocol is built from byte-serialised integers, so a byte-order assumption would surface as a differing region root rather than as a compile error — its passing shows the serialisation commits explicit byte order rather than reinterpreting memory.

ARM64 is confirmed on silicon rather than only under emulation: CI runs the same vectors on native ARM64 Linux and on Apple Silicon on every push, and both reproduce the reference limbs byte for byte. The remaining emulated target is s390x, kept because big-endian is where a byte-order assumption would surface, not because anyone expects to run a miner there. What is still untested is a second compiler version.

4 Measured Substrate Results

Each result below is produced by a standalone gate in the repository, is bit-reproducible, and is paired with a control that removes the proposed mechanism and shows the behaviour collapse. Where the substrate cannot do something, the gate measures the limit instead of hiding it. The claim is narrow and is made the same way throughout: this is evidence that the architecture *permits* structured behaviour, not a claim of intelligence.

4.1 Memory and generalisation

Clean attractor capacity is 8 patterns at N=48, at or above the Hopfield reference of ≈7 for this size. A covariance write rule lifts recall of overlapping patterns at K=8 from 10% to 52%. With scale, storage and generalisation rise together: storage at K=8 from 25% to 79%, and recall of an unseen pro-

totype from its exemplars from 33% to 72%, across N=48 . . . 192. The controls are the comparison itself — the plain rule and the small network. This is associative memory that abstracts, not a lookup table.

4.2 Sequence and context

Memory is symmetric; order is not. A directed transition operator carries the state along a learned chain, and the controls establish the order is genuine: the trajectory goes forward and not back, two sequences coexist without interference, and traversal scales with N (55%→80%). Lifted into the dynamics, a cyclic five-pattern sequence seeded once walks itself for 16 correct steps — more than three full loops — where switching off the delayed asymmetric drive gives a longest run of 1.

Where two sequences cross at a shared state, pairwise traversal from the current state alone is at chance (50%); carrying one step of path context resolves it to 100%. With two shared states in a row, one step of context fails (50%) and two steps resolve it — context depth must match the depth of the ambiguity. Given a ceiling, the traverser *chooses* that depth rather than being told it.

4.3 Analogy, and analogy as inference

gate	result	control
functor, given pairs	100% OOD	0%
correspondence discovered	10/10 exact	10%
alignment earned	100% OOD	4%
discovery + map fused	97%	30%
partial overlap	89%/92%	67% prec.
generation	**83%**	**7%**

Table 2: The analogy arc, from a supplied correspondence to one discovered jointly with the mapping, and finally to inference. Controls are the non-additive map, surface matching, the no-functor condition, and the identity correspondence.

The arc walks from the most given setting to the least. Interpolating from an unaligned to an aligned configuration reproduces the falling Dirichlet energy (168→42) and rising out-of-distribution accuracy (0%→100%) *together*. With the correspondence hidden behind an unknown permutation and no shared surface, iterative refinement of relational signatures recovers it exactly, while surface matching stays at chance — and a structurally regular graph, where every entity has the same outgoing label set, is correctly reported as unrecoverable. A single EM-style settling then recovers correspondence and mapping at once (97%). At the capstone, a target category is only partially observed, and relations never seen are generated correctly 83% of the time, degrading gracefully as more is hidden and staying far above both baselines.

4.4 Binding by phase, and forgetting

An associative store superposes everything active into one state, so several co-active objects blur. With a deterministic CORDIC trigonometry certified in the fixed-point module, objects are tagged at evenly spaced phases and read back by projection: three and four overlapping objects sharing cells are each recovered at 100% precision and recall, where collapsing them to one phase gives 44% — the superposition catastrophe. The binding survives the full relaxation rather than only the readout, and corrupted phases within an assembly heal back to coherence, while a desynchronising coupling makes separation between assemblies an attractor (19° →80°, 87% demultiplexing).

Forgetting is equally exact. Resonance decays as a certified fixed-point exponential, is refreshed by real retrieval, and is pruned below a floor: on a fixed access schedule the used concepts survive and the never-used are pruned at exactly the epoch the closed form predicts, the iterated decay matching it to 2.3×10^{-10}. A pruned concept is structurally gone — it no longer completes from a partial cue (0% against 100% for a survivor) — and the no-decay control keeps everything.

That same resonance is what the economy uses to decide forgetting, so the substrate's forgetting and the economy's forgetting are one event rather than two models running side by side.

5 The Coordination Layer, and What Is Not Yet True

The protocol layer is built: a Φ ledger with breathing supply (tail issuance plus fee and prune burn), stake-weighted sortition with quorum finality, an independent state-root recomputation that rejects an invalid block, on-chain governance of the validator set and economic parameters, height-gated protocol versioning with upgrade signalling, trustless catch-up sync that re-verifies every block, peer discovery by exchange, and a coordinator-free proposer loop. Forty gates cover this layer specifically (62 across the repository), and every value-moving operation carries the actor's signature, the chain id and a nonce, with mints additionally requiring an episode

receipt that the ledger re-verifies.

That last property is recent, and the way it was missing is instructive. Four operations — including the mint — were originally defined as always-valid: the proposer folded them into the state root, every honest validator recomputed the same root and signed, and the recomputation-based dispute check passed. That check proves a block was applied *consistently*; it can never prove an operation was *authorized*. No amount of consensus rigour catches an operation that is valid by definition, which is why the fix belonged in the operation grammar and not in consensus. Twenty-seven negative tests now assert that the unauthorized paths are refused, and CI makes the functional suite depend on them.

We state the boundary explicitly:

- **No network has run.** Both daemons bind loopback only, and the consensus daemon does not persist state. Everything above is verified in-process or over local sockets. A live multi-machine deployment has never been attempted.
- **Nothing is economically at stake.** Validator stake is a configuration value rather than bonded Φ; slashing has a verified primitive but no on-chain path. There is no fork choice and no view change.
- **Minting is authorized, not decentralized.** Episode receipts are signed by a single authority key. Replacing it with validator-side rederivation of the episode identity — which is what would finally make the descent *measured* rather than asserted — is the next step on that axis.
- **The economy is simulated.** Breathing supply is demonstrated in a closed simulation and in gated ledger paths, not against real demand.

None of this is currently dangerous, precisely because nothing is exposed and no value is attached. It becomes dangerous the moment that ordering is reversed.

6 Where This Goes

Two strands proceed in parallel and meet when both are green.

The protocol strand is ordinary distributed-systems engineering with a defined sequence: durable state and a content-addressed genesis descriptor first, since almost everything depends on a node that survives a restart; then ledger-backed validator staking, because slashing without capital at risk is theatre; then block timestamps, fork choice and view change; then transport hardening immediately before a port is opened, and not before. The acceptance test is unchanged and falsifiable: a party we do not control installs the protocol, joins, verifiably syncs a region, serves a verified episode, and earns Φ into their own wallet without anyone's permission.

The capability strand is where the open-weight network is assembled, and its target is settled even though none of it is built. The results in §4 show what an energy-based associative graph permits; they do not show that world-scale semantics emerge from it unaided, and we do not claim they do — that hope is withdrawn as a product assumption. A fixed open-weight model is bound at the protocol boundary as the reasoning layer, supplying semantics rather than waiting for them, while the graph remains the memory and knowledge layer that associates, retains and forgets.

Four constraints keep this an addition rather than a redefinition. The model runs *outside* the certified loop, at the same I/O boundary where fixed-point conversion already happens, so consensus verifies the deterministic graph transition and never the model's arithmetic — bit-exact floating-point inference across heterogeneous hardware is not achievable and is not attempted. Its weights are a protocol constant pinned by content hash, so a change is a coordinated fork rather than a silent divergence. Only permissively licensed weights qualify: an acceptable-use policy, an attribution or pass-through obligation, a user-count cap, or a vendor's right to restrict usage remotely is disqualifying on mechanics, because an anonymous host cannot satisfy an obligation to a party it never contracts with.

The fourth is the rule everything else rests on: *inference mints nothing.* Running a model is a fee-paid service that anyone can also run locally; only *retained memory* mints, and forgetting burns. If that rule ever slips, the protocol collapses into a commodity inference market with extra steps, and its reason to exist goes with it.

That framing also identifies the moat honestly. A fixed open-weight model is free and reproducible by anyone; a shared, permissionless, self-pruning memory whose economics are enforced by a certified engine is not. The scarce object is the retained graph, not the arithmetic that reads it.

7 Application Surface

Shared long-term memory for agent systems. A read/write associative store with an explicit retention economy: what is used stays, what is not is forgotten and costs its stake. Complementary to large generative models rather than competing with them.

Verifiable computation as a service. Episodes whose honesty is established by recomputation against a bit-exact contract, not by trusting a scorer — a property that decentralized inference markets do not currently offer.

Self-limiting canonical state. A chain whose stored state is bounded by an intrinsic metabolism rather than by pricing policy.

The engine on its own. The certified Layer-0 core — deterministic associative memory with binding by phase and exact forgetting, in integer arithmetic on commodity hardware — is usable independently of any network, wherever reproducibility and inspectability are requirements rather than preferences.

8 Conclusion

AGN proposes that the two hard problems of decentralized AI — proving work was honest, and keeping shared state bounded — are answered by the same substrate rather than by two bolted-on mechanisms. The proof of work is the energy law the substrate already obeys; the bound on state is the forgetting it already performs.

What is demonstrated today is the memory half and its arithmetic: an associative graph that stores and generalises, carries order and conditional structure, discovers analogies and uses them to infer, binds several objects at once without blurring them, and forgets deterministically — all in integer arithmetic that reproduces byte-identically across five dissimilar targets, four of them on native hardware. What is built but not yet proven is the network around it. What is designed and not yet built is the reasoning layer above it: a fixed open-weight model that reads from and writes to this memory, earning fees and never minting. We have said exactly where each of those lines falls.

The next milestone is not a larger claim. It is two machines.